

# TTYNAME

Can return a non-null-terminated string.

Sean Barnum, Digital, Inc. [vita<sup>1</sup>]

Copyright © 2007 Digital, Inc.

2007-04-23

## Part "Original Digital Coding Rule in XML"

Mime-type: text/xml, size: 4158 bytes

<b>Attack Category</b>	• Path spoofing or confusion problem				
<b>Vulnerability Category</b>	• No Null Termination • Buffer Overflow				
<b>Software Context</b>	• File Path Management				
<b>Location</b>	• unistd.h				
<b>Description</b>	<p>The ttynname(int fildes) function returns a pointer to a string containing the path name of the terminal device associated with file descriptor fildes. The return value may point to static data, possibly overwritten by the next call.</p> <p>It is possible for this function, depending on call profile and implementation to return a non-null-terminated string.</p>				
<b>APIs</b>	<b>Function Name</b>	<b>Comments</b>			
	ttynname				
<b>Method of Attack</b>					
<b>Exception Criteria</b>					
<b>Solutions</b>	<b>Solution Applicability</b>	<b>Solution Description</b>	<b>Solution Efficacy</b>		
	Always	Ensure that the ttynname string is null terminated before use.	Effective.		
<b>Signature Details</b>	char *ttynname ( int desc );				
<b>Examples of Incorrect Code</b>	<pre>#include &lt;stdio.h&gt; #define ENUF(msg, value) {perror(msg); exit(value); } main(argc, argv) int argc; char *argv[]; { int i, pid, fd[2];</pre>				

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

```

char *gets(), *ttynname();
char tty[4], *ptr;

if ( pipe(fd) != 0 ) ENUF("pipe
creation error", 1);
i = 0;
while ( (pid=fork()) == -1) /*
try a few times before giving up
*/
if (++i > 5 ) ENUF("failed to
form after 5 attempts", 2) else
sleep(5);

if (pid == 0) { /* CHILD process
*/
ptr = ttynname(fileno(stdin));
tty[0] = ' ' ; tty[1] =
ptr[strlen(ptr)-2];
tty[2] = ptr[strlen(ptr)-1];
tty[3] = ' ' ; tty[4] = '\0';

fprintf(stdout, " Child pipe fds
before dup: fd[0] %d, fd[1] %d
\n",fd[0],fd[1]);
close(fd[1]); /* not using pipe
write fd=4 */
close(0); /* redirecting stdin
fd=0 */
i = dup(fd[0]); /* duplicate pipe
read with stdin i.e. 0=3 */
/* ... */

```

#### **Examples of Corrected Code**

```

//just use ttynname_r to have a
thread-safe version

status = ttynname_r (0, stdin_str,
sizeof (stdin_str));
if (status != 0)
err_abort (status, "Get stdin");

```

#### **Source References**

- [Rough Auditing Tool for Security \(RATS\)<sup>2</sup>](#)
- <http://man.he.net/man3/ttynname>

#### **Recommended Resource**

#### **Discriminant Set**

<b>Operating Systems</b>	<ul style="list-style-type: none"> <li>• UNIX</li> <li>• Windows</li> </ul>
<b>Languages</b>	<ul style="list-style-type: none"> <li>• C</li> <li>• C++</li> </ul>

## **Cigital, Inc. Copyright**

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Digital, including information about “Fair Use,” contact Digital at [copyright@digital.com](mailto:copyright@digital.com)<sup>1</sup>.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

---

1. <mailto:copyright@digital.com>